Note
Before using this information and the product it supports, read the information in "Notices" on page 25.
This edition applies to version 10, release 1 of IBM Spectrum LSF (product number 5725G82) and to all subsequent releases and modifications until otherwise indicated in new editions.
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LSF resource connector auditing</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>rclogsvalidate</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>LSF_RC_AUDIT_LOG</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>RC_MAX_AUDIT_LOG_KEEP_TIME</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>RC_MAX_AUDIT_LOG_SIZE</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>lsportcheck utility</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>UNIX support for Ismail</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>LSB_MAILPROG</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>LSB_MAILSERVER</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Interactive logon</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>About interactive logon</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Scope</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>LSB_LOGON_INTERACTIVE</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Enable interactive logon</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Automatically create the LSF hosts file on Linux/UNIX</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>hostsfilesetup</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Chapter 6. Use lsreghost instead of lsreglocalhost</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Chapter 7. Configure LSF to use TCP for communication</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>LSF_CALL_LIM_WITH_TCP</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>LSF_ANNOUNCE_MASTER_TCP_WAITTIME</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Chapter 8. Enhance LSF security with eauth.cve</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Chapter 9. Configure LSF to use Amazon Web Server (AWS) as a resource provider</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Chapter 10. Configuring health check script for high availability applications</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Using a Job Monitor to update activity states</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Using the Job Monitor protocol</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Notices</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Trademarks</td>
<td>27</td>
</tr>
</tbody>
</table>
Chapter 1. LSF resource connector auditing

With this release, LSF will log resource connector VM events along with usage information into a new file `rc.audit.x` (one log entry per line in JSON format). The purpose of the `rc.audit.x` log file is to provide evidence to support auditing and usage accounting as supplementary data to third party cloud provider logs. The information is readable by the end user as text and is hash protected for security.

LSF also provides a new command-line tool `rclogsvalidate` to validate the logs described above. If the audit file is tampered with, the tool will identify the line which was modified and incorrect.

New parameters have been added to LSF in the `lsf.conf` configuration file:

- `LSF_RC_AUDIT_LOG`: If set to `Y`, enables the resource connector auditor to generate log files.
- `RC_MAX_AUDIT_LOG_SIZE`: An integer to determine the maximum size of the `rc.audit.x` log file, in MB.
- `RC_MAX_AUDIT_LOG_KEEP_TIME`: An integer that specifies the amount of time that the resource connector audit logs are kept, in months.

**rclogsvalidate**

Displays audit logs for LSF resource connector.

**Synopsis**

```
rclogsvalidate -l [-d start_time,end_time]
rclogsvalidate -n number
rclogsvalidate [-h] [-V]
```

**Description**

**Options**

- `-l [-d start_time,end_time]`
  Displays audit logs for the last hour.
  To display audit logs for a specified time interval, use the `-d` option and specify the times in the following format: `yyyy/mm/dd/HH:MM`
  Use a comma to separate the start and end times and do not use spaces when specifying the time interval.

- `-n number`
  Displays and checks audit logs in consecutive order starting with the specified number (`rc.audit.number`) and ending with the current log. You must specify a positive integer.
  If none of the checked logs is tampered, this command exits with return code 0.
  If any logs are tampered, this command stops and exits with return code 99.
This command returns an error if the rc.audit.integer file (from rc.audit.number to the current log file) does not exist.

- h  Prints command usage to stderr and exits.
- V  Prints LSF resource connector release version to stderr and exits.

### LSF_RC_AUDIT_LOG

**Syntax**

```
LSF_RC_AUDIT_LOG=Y | y | N | n
```

**Description**

If set to Y or y, enables the LSF resource connector auditor to generate log files.

**Default**

N.

**See also**

RC_MAX_AUDIT_LOG_SIZE, RC_MAX_AUDIT_LOG_KEEP_TIME

### RC_MAX_AUDIT_LOG_KEEP_TIME

**Syntax**

```
RC_MAX_AUDIT_LOG_KEEP_TIME=integer
```

**Description**

Specifies the amount of time that LSF keeps the resource connector audit log files, in months.

**Default**

Not defined.

**See also**

LSF_RC_AUDIT_LOG, RC_MAX_AUDIT_LOG_SIZE

### RC_MAX_AUDIT_LOG_SIZE

**Syntax**

```
RC_MAX_AUDIT_LOG_SIZE=integer
```

**Description**

Specifies the maximum size of the LSF resource connector audit log files, in MB.

**Default**

Not defined.
See also

LSF_RC_AUDIT_LOG, RC_MAX_AUDIT_LOG_KEEP_TIME
Chapter 2. Isportcheck utility

A new Isportcheck utility has been added to LSF. This utility can be used to check the required ports for LSF and include detailed information, whether it is being used or not.

The Isportcheck utility only checks ports on the host for availability. It discovers the ports by reading the configuration files. If the line is commented out or if there is no value, it will use the default values.

The Isportcheck utility must be executed by the root user, since the tool uses 'netstat' and needs root to get the complete information on the ports of the OS.

Before running this tool, you must source the profile or set the environment variable LSF_TOP.

The utility is installed at <LSF_TOP>/<VERSION>/<PLATFORM>/bin/, for example, /opt/lsf/10.1/linux2.6-glibc2.3-x86_64/bin/

Usage:

lsportcheck

lsportcheck -h

lsportcheck -l|-m | -s

Description:

Without arguments will output command usage and exit.

-h Output command usage and exit.

-l List TCP and UDP ports on master.

-l -m List TCP and UDP ports on master.

-l -s List TCP and UDP ports on slave.

Note: Isportcheck can only be run by root.

Source the relative IBM Spectrum LSF shell script after installation:

For csh or tcsh: 'source $LSF_ENVDIR/cshrc.lsf'

For sh, ksh, or bash: 'source $LSF_ENVDIR/profile.lsf'

Example output:

Example of the output using command Isportcheck -l or Isportcheck -l -m on LSF master:
Checking ports required on host [mymaster1]

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Port Number</th>
<th>Protocol</th>
<th>Binding Address</th>
<th>PID/Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>lim</td>
<td>7869</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1847</td>
</tr>
<tr>
<td>lim</td>
<td>7869</td>
<td>UDP</td>
<td>0.0.0.0</td>
<td>1847</td>
</tr>
<tr>
<td>res</td>
<td>6878</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1881</td>
</tr>
<tr>
<td>sbatchd</td>
<td>6882</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1890</td>
</tr>
<tr>
<td>mbatchd</td>
<td>6881</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1921</td>
</tr>
<tr>
<td>mbatchd</td>
<td>6891</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1921</td>
</tr>
<tr>
<td>pem</td>
<td>7871</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1879</td>
</tr>
<tr>
<td>vemkd</td>
<td>7870</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1880</td>
</tr>
<tr>
<td>egosc</td>
<td>7872</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>3226</td>
</tr>
</tbody>
</table>

Optional ports:

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Port Number</th>
<th>Protocol</th>
<th>Binding Address</th>
<th>PID/Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsgserver</td>
<td>9090</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>[Not used]</td>
</tr>
<tr>
<td>named</td>
<td>53</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>[Not used]</td>
</tr>
<tr>
<td>named</td>
<td>53</td>
<td>UDP</td>
<td>0.0.0.0</td>
<td>[Not used]</td>
</tr>
<tr>
<td>named</td>
<td>953</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>[In use by another program]</td>
</tr>
</tbody>
</table>

Example output:

Example of the output using command `lsportcheck -l -s` on LSF slave:

Checking ports required on host [host1]

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Port Number</th>
<th>Protocol</th>
<th>Binding Address</th>
<th>PID/Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>lim</td>
<td>7869</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1847</td>
</tr>
<tr>
<td>lim</td>
<td>7869</td>
<td>UDP</td>
<td>0.0.0.0</td>
<td>1847</td>
</tr>
<tr>
<td>res</td>
<td>6878</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1881</td>
</tr>
<tr>
<td>sbatchd</td>
<td>6882</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1890</td>
</tr>
<tr>
<td>pem</td>
<td>7871</td>
<td>TCP</td>
<td>0.0.0.0</td>
<td>1879</td>
</tr>
</tbody>
</table>
Chapter 3. UNIX support for Ismail

The mail program Ismail is now available for LSF on UNIX platforms, and works like Ismail.exe for LSF on Windows.

LSB_MAILSERVER and LSB_MAILPROG in lsf.conf have been updated to provide support for Ismail on UNIX. Note the configuration differences between UNIX and Windows.

**LSB_MAILPROG**

**Syntax**

LSB_MAILPROG=file_name

**Description**

Path and file name of the mail program used by LSF to send email. This is the electronic mail program that LSF uses to send system messages to the user. When LSF needs to send email to users it invokes the program defined by LSB_MAILPROG in lsf.conf.

In a mixed cluster, you can specify different programs for Windows and UNIX. You can set this parameter during installation on Windows.

LSF provides the sendmail program, which supports the sendmail protocol on UNIX. You can also write your own custom mail program and set LSB_MAILPROG to the path where this program is stored.

LSF also provides the Ismail program for Unix (supporting SMTP) and the Ismail.exe program for Windows (supporting both SMTP and Microsoft Exchange Server). If Ismail is specified, the parameter LSB_MAILSERVER must also be specified. On Windows, Ismail.exe can be configured directly. On Unix, the full path to the Ismail binary is required for configuration.

If you change your mail program, the LSF administrator must restart sbatchd on all hosts to retrieve the new value.

**UNIX**

By default, LSF uses /usr/lib/sendmail to send email to users. LSF calls LSB_MAILPROG with two arguments; one argument gives the full name of the sender, and the other argument gives the return address for mail.

**LSB_MAILPROG** must read the body of the mail message from the standard input. The end of the message is marked by end-of-file. Any program or shell script that accepts the arguments and input, and delivers the mail correctly, can be used.

**LSB_MAILPROG** must be executable by any user.

**Windows**

If LSB_MAILPROG is not defined, no email is sent.
Examples
LSB_MAILPROG=lsmail.exe
LSB_MAILPROG=/serverA/tools/lsf/bin/unixhost.exe

Default
Unix: /usr/lib/sendmail
Windows: empty

See also
LSB_MAILSERVER, LSB_MAILTO

LSB_MAILSERVER

Syntax
For Windows: LSB_MAILSERVER=mail_protocol:hostname
For UNIX: LSB_MAILSERVER=hostname

Description
Part of mail configuration on Windows and Unix.
This parameter only applies when lsmail is used as the mail program (LSB_MAILPROG=lsmail.exe). Otherwise, it is ignored.
On Windows, specify the protocol and name of the mail server host. For an SMTP mail host, specify SMTP:hostname. For an exchange mail host, specify EXCHANGE:hostname. That is:
LSB_MAILSERVER=[SMTP: | EXCHANGE:]hostname
On UNIX, specify just the name of the mail server host. That is:
LSB_MAILSERVER=hostname

Note: On Unix, only SMTP is supported as the mail_protocol.
This parameter is set during installation of LSF or is modified by the LSF administrator.
If this parameter is modified, the LSF administrator must restart sbatchd on all hosts to retrieve the new value.

Default
Not defined

See also
LSB_LOCALDIR, LSB_MAILTO, and LSB_MAILPROG
Chapter 4. Interactive logon

Use the interactive logon option to modify the Windows Logon Type on the execution host (the host running the job) from **logon as a batch job** to **logon interactively** when jobs are run on the execution host.

About interactive logon

In order for jobs to run, the Windows user account privilege **logon as a batch job** must be set on all Windows hosts that can run jobs for Windows submission user accounts. A job fails if the submitting Windows user account does not have this privilege on the execution host.

Enabling interactive logon is an alternative to setting the **logon as a batch job** privilege for all users running work on Windows hosts.

In environments where it is not desirable to assign a **logon as a batch job** privilege to all Windows user accounts, you can enable the interactive logon option and assign the **logon interactively** privilege to all Windows user accounts instead. LSF uses **logon interactively** instead of **logon as a batch job** when running jobs.

Scope

<table>
<thead>
<tr>
<th>Operating system</th>
<th>• Windows hosts (all supported Platforms).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not required for</td>
<td>• UNIX execution hosts.</td>
</tr>
<tr>
<td></td>
<td>• User account mapping UNIX hosts.</td>
</tr>
<tr>
<td>Application</td>
<td>• This option applies to Windows execution hosts.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>• On Windows hosts that may run work, you must grant the <strong>logon interactively</strong> privilege to Windows user accounts.</td>
</tr>
<tr>
<td>Limitations</td>
<td>• You can configure this option and use <strong>logon interactively</strong> or use <strong>logon as a batch job</strong>. You cannot use both at the same time.</td>
</tr>
</tbody>
</table>

LSB_LOGON_INTERACTIVE

Syntax

```
LSB_LOGON_INTERACTIVE=Y|y|N|n
```

Description

If **LSB_LOGON_INTERACTIVE** is set to **Y** or **y** in **lsf.conf**, LSF executes jobs on the Windows host with the **logon interactively** Windows Logon Type.

If **LSB_LOGON_INTERACTIVE** is set to **N**, **n**, or any other value in **lsf.conf**, LSF executes jobs on the Windows host with the **logon as a batch job** Windows Logon Type.
If LSB_LOGON_INTERACTIVE is not set in lsf.conf, LSF executes jobs on the Windows host:

- With the **logon as a batch job** Windows Logon Type when the operating system on the Windows execution host is earlier than Vista.
- With the **logon interactively** Windows Logon Type when the operating system on the Windows execution host is Vista or later.

**Default**

LSB_LOGON_INTERACTIVE=N, LSF executes jobs on the Windows host with the **logon as a batch job** Windows Logon Type.

### Enable interactive logon

#### Before you begin

**Prerequisites:**

Set the **logon interactively** privilege for submission Windows user accounts on all hosts where a job submitted by these Windows users could run.

**Procedure**

1. As the LSF administrator, set LSB_LOGON_INTERACTIVE=Y in lsf.conf.
   This setting authorizes Windows user accounts that have the **logon interactively** privilege set in the local security policy on the host that runs the job.

   **Important:**

   The job will fail if the submission user account does not have the **logon interactively** privilege on the Windows execution host.

2. Run `badmin reconfig`. 

---

Readme
Chapter 5. Automatically create the LSF hosts file on Linux/UNIX

About this task

The LSF hosts file located in $LSF_CONFDIR associates host names and IP addresses in LSF. This is useful for hosts with multiple IP addresses and different official host names configured at the system level. You can use the command hostsfilesetup to automatically create the LSF hosts file for all hosts in your cluster, and to update the LSF hosts file.

Procedure

1. Install LSF and set up the lsf.cluster.cluster_name file with the official name of hosts.
2. As root or the LSF primary administrator, log on to one of the master candidate hosts and use hostsfilesetup to automatically create the $LSF_CONFDIR/hosts file.

hostsfilesetup

Note: The LSF primary administrator or root must be able to access all UNIX/Linux hosts in the cluster with password-less rsh or ssh from the host on which this command is run. If password-less rsh is not configured, access to remote hosts fails. If password-less ssh is not configured, the user account running this command is prompted for a password.

hostsfilesetup

Synopsis

hostsfilesetup [ -m host_name [ , host_name , ...] | -m all ][ -v ipv4 | -v ipv6 ][ -r rsh | -r ssh ]

hostsfilesetup -h | -V

Description

You must be root or the primary LSF administrator to run this command have access to all UNIX/Linux hosts in the cluster with password-less rsh or ssh from the host on which this command is run. If password-less rsh is not configured, access to remote hosts fails. If password-less ssh is not configured, the user account running this command is prompted for a password.

This command is only available on Linux/UNIX from LSF server hosts that have access to the lsf.cluster.cluster_name file, and from LSF master candidate hosts.

By default, when no options are used, creates the LSF hosts file($LSF_CONFDIR/hosts) and maps host names to IP addresses and host aliases, for all hosts in the cluster defined in the lsf.cluster.cluster_name file. By default, uses the shell defined by the parameter LSF_RSH in the lsf.conf file to log in to each host to
discover IP address and alias information. If LSF_RSH is not defined and a shell is not specified, rsh is used. IP addresses are written in dotted quad notation (IPv4) in the $LSF_CONFDIR/hosts file.

The $LSF_CONFDIR/hosts file has the following format:

```
ip_address official_host_name [alias [alias ...]]
```

For example, host1.example.com has two network interfaces and would be listed in the hosts file as:

```
190.123.55.77 host1.example.com alias-AA
190.123.55.88 host1.example.com alias-BB
```

**Note:** In certain cases, `hostsfilesetup` may not be able to retrieve all aliases for a host. If any host aliases are missing, you manually add them to the hosts file.

When a new file is created, the file is owned by the primary LSF administrator. If the hosts file exists, original file ownership permissions are maintained.

**Options**

```
-m host_name [, host_name , ...] -m all
```

Updates the $LSF_CONFDIR/hosts file and adds entries to map IP addresses to host names and aliases for the specified hosts. The host names that you specify must be the same names as listed in the `lsf.cluster.cluster_name` file.

If the keyword all is used, updates entries for all hosts in the `lsf.cluster.cluster_name` file.

```
-v ipv4 | -v ipv6
```

Specifies whether entries in the hosts file are written in a dotted quad notation (IPv4) or IP Next Generation (IPv6) format.

If you specify ipv6, ensure `LSF_ENABLE_SUPPORT_IPV6=y` in the configuration file `lsf.conf`.

```
-r rsh | -r ssh
```

Indicates whether to use rsh or ssh to log on to the specified hosts to discover host IP and alias information.

```
-h
```

Prints the command usage to stderr and exits.

```
-V
```

Prints the product release version to stderr and exits.

**Examples**

Create the `$LSF_CONFDIR/hosts` file, include all hosts in the cluster, and use IPv4 format to write to the file and rsh to log in to each host to discover IP address and aliases:

```
$ hostsfilesetup -r rsh
```

Update the `$LSF_CONFDIR/hosts` file: hosts host1.example.com, host2.example.com, IPv4 format to write to the file, rsh to log in to each host to discover IP address and aliases:

```
$ hostsfilesetup -m host1.example.com,host2.example.com -r rsh
```
Update the entire $LSF_CONFDIR/hosts file, include all hosts in the cluster, and use IPv4 format and ssh:

$ hostsfilesetup -m all -r ssh

Overwrite the existing $LSF_CONFDIR/hosts file, include all hosts in the cluster, and use IPv6 format and ssh:

$ hostsfilesetup -v ipv6 -r ssh
Chapter 6. Use lsreghost instead of lsreglocalhost

In version 10.1, IBM Spectrum LSF introduced the command `lsreghost`. This command directly registers LSF host names and IP addresses with LSF servers so that LSF servers can internally resolve these hosts without requiring a DNS server.

As a result, the command `lsreglocalhost` is now deprecated. Use `lsreghost` instead. The `lsreghost` command resolves security vulnerabilities in `lsreglocalhost`. The `lsreglocalhost` command allows non LSF hosts to register as valid client hosts due to the fact that the command can be run by anyone on any client host when the IP range is matched.
Chapter 7. Configure LSF to use TCP for communication

LSF uses a mix of UDP and TCP calls for communication. If UDP is not reliable in your environment, configure LSF to use only TCP to communicate. Configure the parameters LSF_CALL_LIM_WITH_TCP and LSF_ANNOUNCE_MASTER_TCP_WAITTIME in the configuration file lsf.conf.

### LSF_CALL_LIM_WITH_TCP

**Syntax**

```
LSF_CALL_LIM_WITH_TCP=Y|y|N|n
```

**Description**

Use this parameter when UDP is not reliable in your environment. Defines whether LSF uses TCP or UDP to communicate with lims in the cluster.

When set to y, LSF uses only TCP to communicate with lims in the cluster.

When set to n, LSF uses UDP to communicate with lims in the cluster.

**Note:** Clients and daemons may still use TCP to communicate with the master lim when LSF_CALL_LIM_WITH_TCP=n, as LSF uses a mix of TCP and UDP calls to communicate.

If you are using resource connector to borrow hosts from Amazon Web Services (AWS), set LSF_CALL_LIM_WITH_TCP=y to provide a reliable protocol for dynamic hosts to connect to LSF.

If you are using LSF with a non-shared file system and the parameter LSF_GET_CONF=lim in the configuration file lsf.conf, you must add LSF_CALL_LIM_WITH_TCP=y in the slave lsf.conf file so that the client also uses TCP when communicating with lim.

**Default**

Undefined or n: LSF uses UDP to communicate with lims in the cluster.

**See also**

LSF_ANNOUNCE_MASTER_TCP_WAITTIME

### LSF_ANNOUNCE_MASTER_TCP_WAITTIME

**Syntax**

```
LSF_ANNOUNCE_MASTER_TCP_WAITTIME=seconds
```

**Description**

Affects communication between the lim on the master host and slave lims. Defines how long a slave lim waits for the master lim to initiate communication. If after
startup the slave lim does not receive any announcement from the master lim within the configured time period, the slave lim sends a request to the master lim.

Important: This parameter can affect performance. Do not use this parameter for large clusters.

Default

60 seconds
Chapter 8. Enhance LSF security with eauth.cve

In LSF, authentication can come by means of external authentication using the LSF `eauth` executable, or by means of identification daemons (`identd`). External authentication provides the highest level of security and is the default method of authentication in LSF. It is installed in the directory specified by the `LSF_SERVERDIR` parameter in the `lsf.conf` file.

By default, `eauth` uses an internal key to encrypt authentication data, but you may use a customized external key to improve security. You can also write your own `eauth` executable to meet the security requirements of your cluster, using the default `eauth` as a demonstration of the `eauth` protocol.

Update the eauth executable file

You can also replace the default `eauth` executable file with the `eauth.cve` executable file, which automatically generates a site-specific internal key by using 128-bit AES encryption. Rename or move the original `eauth` executable file, then rename the `eauth.cve` executable file to `eauth`.

In Windows hosts, replace the `eauth.exe` executable file with `eauth.cve.exe`

The new `eauth` command rejects LSF requests from hosts with a UTC time offset of more than five minutes compared to the LSF server host.

If you are using IBM Spectrum LSF RTM, you must also update to the corresponding new `eauth` executable file for IBM Spectrum LSF RTM.

**Note:** You must replace the executable file on all LSF hosts in the LSF cluster to work together. Otherwise, LSF commands that run on the hosts without the new `eauth` executable file will encounter authentication problems.
Chapter 9. Configure LSF to use Amazon Web Server (AWS) as a resource provider

Use the `aws_enable.sh` script and the `aws_enable.config` file in `$LSF_TOP/10.1/install` to configure LSF to use Amazon Web Server (AWS) as a resource provider through LSF resource connector. You can find instructions for using the script in the `aws_enable.sh` file.

For detailed steps for enabling AWS in LSF resource connector, see the *Configuring IBM Spectrum LSF Resource Connector for AWS* guide.

For detailed steps for enabling AWS in LSF resource connector, see the LSF Knowledge Center [https://www.ibm.com/support/knowledgecenter/SSWRJV_10.1.0/lsf_resource_connector/lsf_rc_update_lsfconfig.html](https://www.ibm.com/support/knowledgecenter/SSWRJV_10.1.0/lsf_resource_connector/lsf_rc_update_lsfconfig.html)
Chapter 10. Configuring health check script for high availability applications

A Job Monitor is a customized script or binary that is written specifically for an application. You can use a Job Monitor to specify an application's state. Job Monitor is also known as health check script.

Using a Job Monitor to update activity states

You can use a Job Monitor to specify an activity's state.

The Job Monitor is responsible for reporting the application's TENTATIVE state, which indicates that an application is running but is not ready, such as when the application is initializing. This state can only be reported by a proprietary script or binary that has the knowledge of the specific application.

The script or binary is written specifically for one type of application and it is defined as a Job Monitor of the application. The Job Monitor reports the state of the application through stdout, which is parsed by EGO.

An application that has the Job Monitor defined will go to the TENTATIVE state first after the application process is started. The Job Monitor then updates the application's state to RUN after initialization. An application without the Job Monitor skips the TENTATIVE state during state transition and reports a RUN state even if the application is not actually ready.

If a Job Monitor is defined for an application, it is started immediately after the application process is started. The following environment variables are set for the Job Monitor process:

- `EGO_ACTIVITY_PID`: Indicates the PID of the application
- `JOB_MONITOR_MAX_UPDATE_INTERVAL`: Specifies the maximum update interval. The Job Monitor should communicate with EGO at least once per update interval

Job Monitor is configured as below in the application definition xml file:

```xml
<ego:ActivitySpecification>
  <ego:JobMonitor>/mypath/mycommand myArg</ego:JobMonitor>
  <ego:JobMonitorMaxUpdateInterval>60</ego:JobMonitorMaxUpdateInterval>
</ego:ActivitySpecification>
```

Using the Job Monitor protocol

Job Monitor passes application state to EGO by printing commands and parameters to stdout.

The following commands are supported:

- `UPDATE_STATE`: Valid parameters for this command are:
  - `READY`
  - `TENTATIVE`
  - `ERROR` (A Job Monitor that reports the ERROR state causes the application to be terminated by EGO.)
- **END**: This command informs EGO that the Job Monitor has nothing more to update and can exit. The application remains in the current state.

**Note**: All parameters must be encapsulated by a pair of single quotes. A single quote cannot be part of the parameter.

**Example on Linux**:

```bash
#!/bin/bash

# Job Monitor Sample

#Check EGO_APPLICATION_PID environment variable
if [ "$EGO_APPLICATION_PID" == "" ];then
  # Failed to get EGO_APPLICATION_PID
  echo "END"
  exit
fi

echo "UPDATE_STATE 'TENTATIVE'"

# Do some initial work, sleep 1 as dummy task
sleep 1

# Set the service instance to ready state which triggers application state to RUN
echo "UPDATE_STATE 'READY'"

# Assume the update interval is 1 second less than max
# If max update interval is not set in service profile, set updateInterval to 1 second
if [ -z $JOB_MONITOR_MAX_UPDATE_INTERVAL ] || [ $JOB_MONITOR_MAX_UPDATE_INTERVAL -le 0 ];then
  let updateInterval=1
else
  let updateInterval=$JOB_MONITOR_MAX_UPDATE_INTERVAL-1
fi

# Monitoring container status
# This example just monitors the PID is existing. Other checks can be done to determine
# if there are issues or not and update status to ERROR. Otherwise it will continue to RUN
while [ 1 ];
    do
        sleep $updateInterval
        # Do extra checking of the service here
        PID=`ps -ef | grep $EGO_APPLICATION_PID | grep -v "grep" | awk '{print $2}';`
        echo "Found PID $EGO_APPLICATION_PID" > /tmp/ego.out
        if [ "$PID" == "" ];then
            echo "UPDATE_STATE 'ERROR'"
            echo "END"
            exit
        fi
    done
```
Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web
sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Intellectual Property Law
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application
programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

IBM, the IBM logo, and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

LSF®, Platform, and Platform Computing are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.